

# ラズパイ倶楽部 ゴールデンウィークのハック報告

(四苦八苦)

2018年5月17日

山本正樹(1号)

(パナソニック AIソリューションセンター)

# 遠隔地の高齢な両親にちょっとメール したいことがある



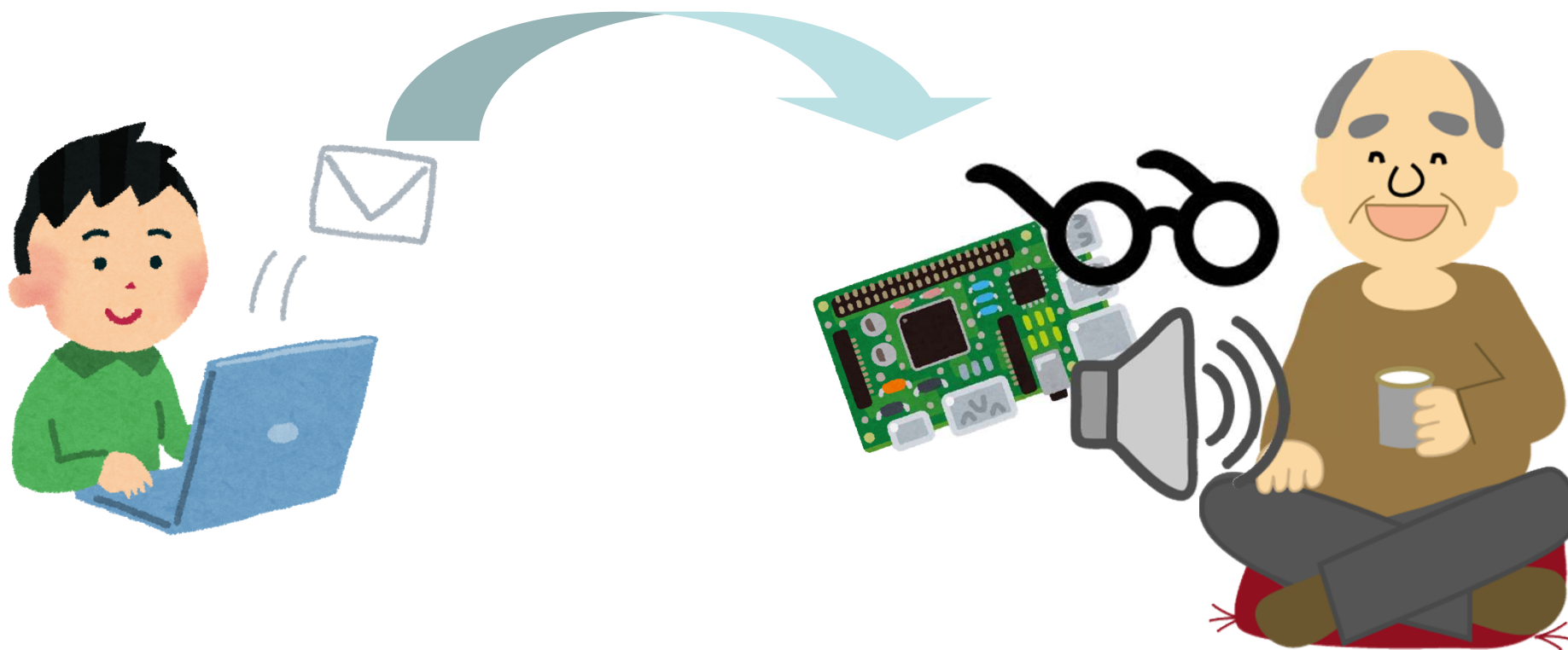
電話したくても、夜遅くなってしまう  
メールで用件伝えられないか？



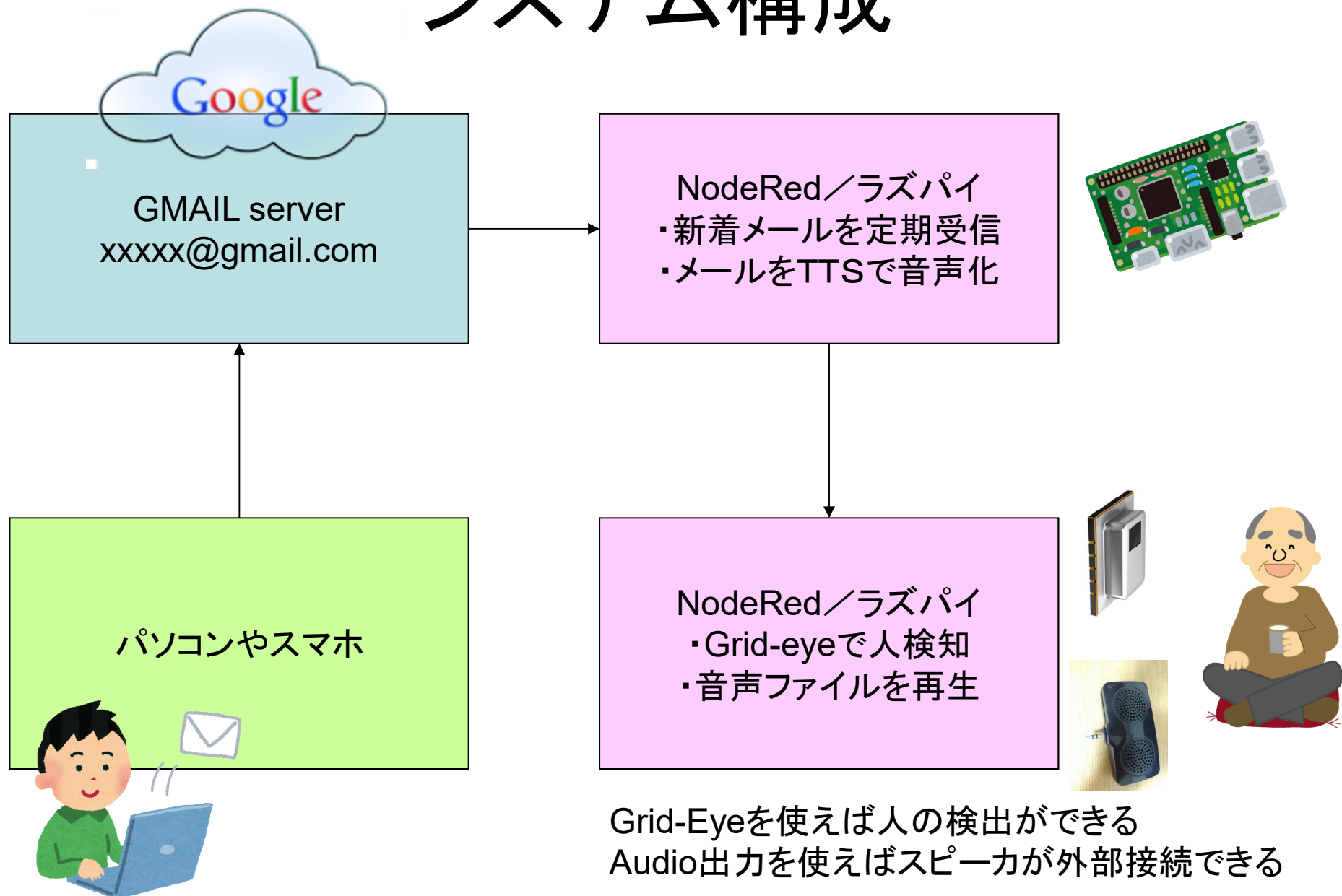
PCは面倒くさいし、信用できない  
そもそも画面の字が見えない

# ラズパイに仲をとってもらいたい

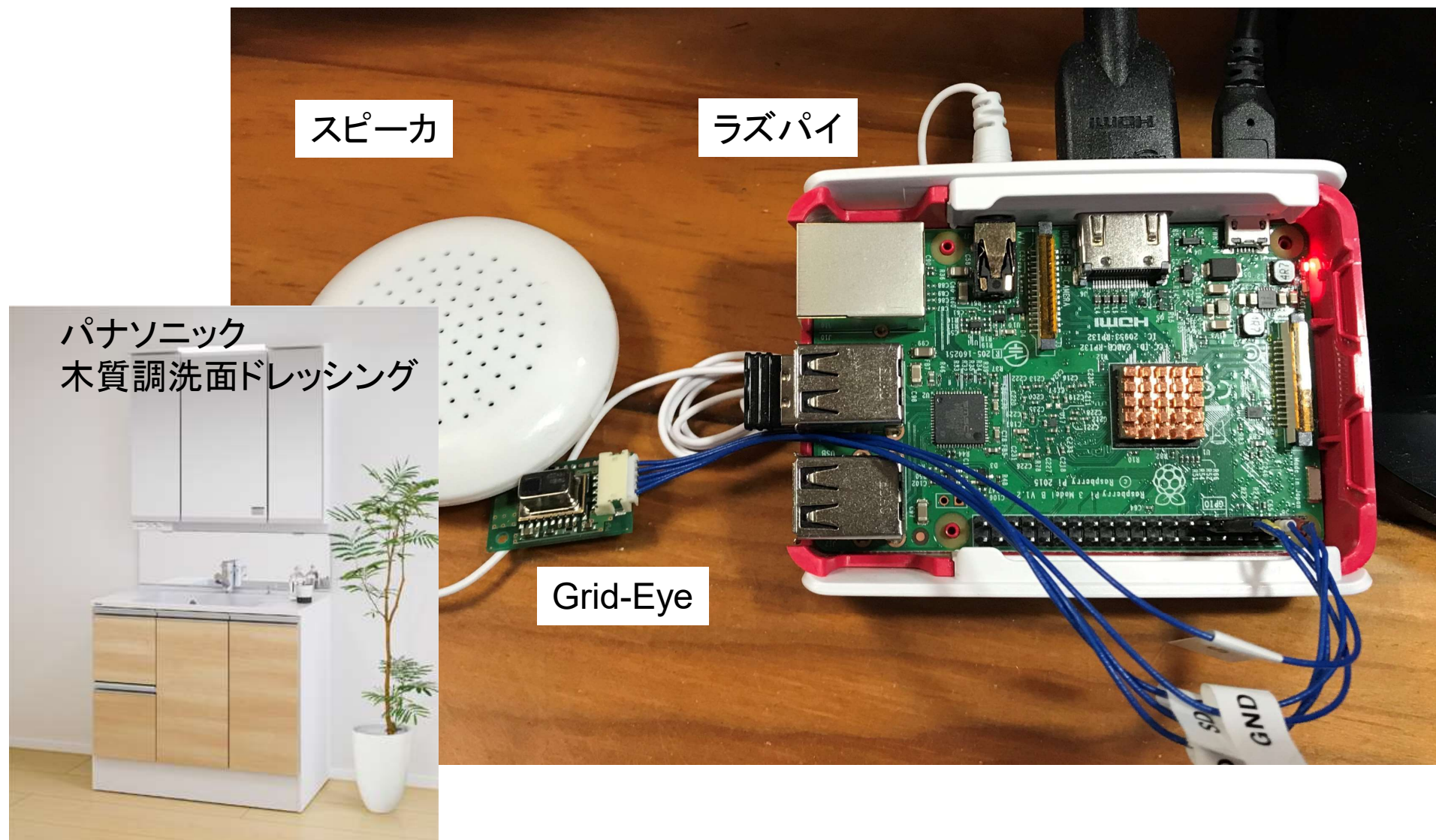
さりげなく見守っていて、TPOを判断してメール読み上げ



# システム構成



# ラズパイ側ハードウェア構成





## 工夫点①

# Nodeの追加 speakerpi



これはNodeRedの追加でできる

これは事前にやっておくこと

イヤホン／HDMIの選択

ボリュームを設定

## node-red-contrib-speakerpi

Readme

4 Dependencies

0 Dependents

## node-red-contrib-speakerpi

A **Node-RED** node to give a provided sound (WAV,OGG) out on the defined output (hdmi or chinch or defined). This node is designed to work on Raspberry Pi and will be good in using with Watson APIs like text-to-speech to demonstrate cognitive iot.

### Install

Run the following command in the root directory of your Node-RED install or home directory (usually ~/.node-red).

```
npm install node-red-contrib-speakerpi
```

Additionally you have to install on the Raspberry Pi

```
sudo apt-get install libasound2-dev
```

**sound set to Analog output**

```
amixer cset numid=3 1
```

**sound set to HDMI**

```
amixer cset numid=3 2
```

**sound set volume**

```
alsamixer
```

## 工夫点②

# Grid-eyeで 人物検出

8x8ピクセルから最高温度を読み取り、  
その値が設定値を超えたら、  
人物検出と判定しプログラム終了

```
pi@masaki: ~/grid_eye
ファイル(F) 編集(E) タブ(T) ヘルプ(H)
33.750
33.750
34.000
34.750
33.500
33.750
33.500
34.250
33.250
34.000
33.750
34.000
33.750
34.500
33.500
34.250
33.750
34.000
34.250
34.250
36.500
Temperature gets high
pi@masaki: ~/grid_eye $ python amg88test2.py 35
```

人がいなくても  
十分温度高い

夏は苦手！

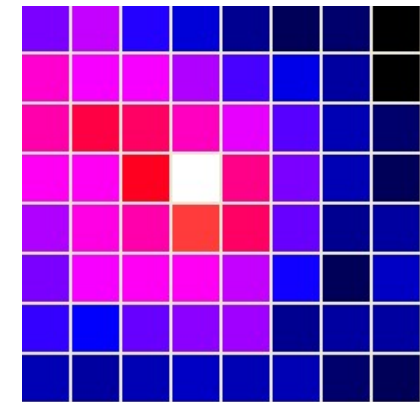
```
amg88test.py x amg88test2.py x
#!/usr/bin/env python
import smbus
import time
import sys
bus = smbus.SMBus(1)
addr = 0x68

def plot(x, minx, maxx, c='*'):
    n = 60 # Max character
    if x < minx:
        x = minx
    if x > maxx:
        x = maxx
    v = int(x/(maxx/n))
    s = c*v+' '*(n-v)
    print s

if len(sys.argv) < 2:
    limit = 34.0
else:
    limit = float(sys.argv[1])
while 1:
    time.sleep(0.1)
    linedata=[]
    for i in range(8) :
        data = bus.read_i2c_block_data(addr , 0x80+0x10*i, 16)
        oneline=[]
        for j in range(8) :
            oneline.append( (int((data[2*j+1] & 0x07) *256 +
            linedata.append(oneline)

            val = 0
            for i, ld in enumerate(linedata):
                for j, ldd in enumerate(ld):
                    if ldd > val:
                        maxi = i
                        maxj = j
                        val = ldd

            #print '%8.3f'%(val)
            if val > limit:
                print 'Temperature gets high', val
                break
```



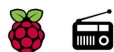
最高温度を求める

## 工夫点③

# Open JTalk

日本語音声合成はインストール  
済みだった

WLOラズパイ倶楽部  
ラズパイでラジオ放送?!



2018年1月25日

<https://qiita.com/wlo>



## Open JTalkを使ってみる

先ほどインストールした、`hts-voice-nitech-jp-atr503-m001`に含まれる音声  
`nitech-jp-atr503-m001`を使って実行してみます。

まず、しゃべらせたい言葉を `voice.txt` に書きます。今回は、以下にします。

`voice.txt`

こんにちは、はじめまして。

その後、コマンドを実行して、音声ファイルを作ります。今回は `test.wav` としま  
す。

```
$ open_jtalk -m /usr/share/hts-voice/nitech-jp-atr503-m001/nitech_jp_atr503_
```

成功すると、`test.wav` というファイルができていますので、

```
$ aplay test.wav
```

でちゃんと再生されれば成功です。

保護された通信 | <https://qiita.com/lutecia16v/items/8d220885082e40ace252>

Qiita コミュニティ キーワードを入力

 @lutecia16v 2016年04月08日に更新

## Raspberry piで日本語音声合成 (Open JTalk)を試してみる。

Raspberry OpenJTalk

👍 84

⚠️ この記事は最終更新日から1年以上が経過しています。

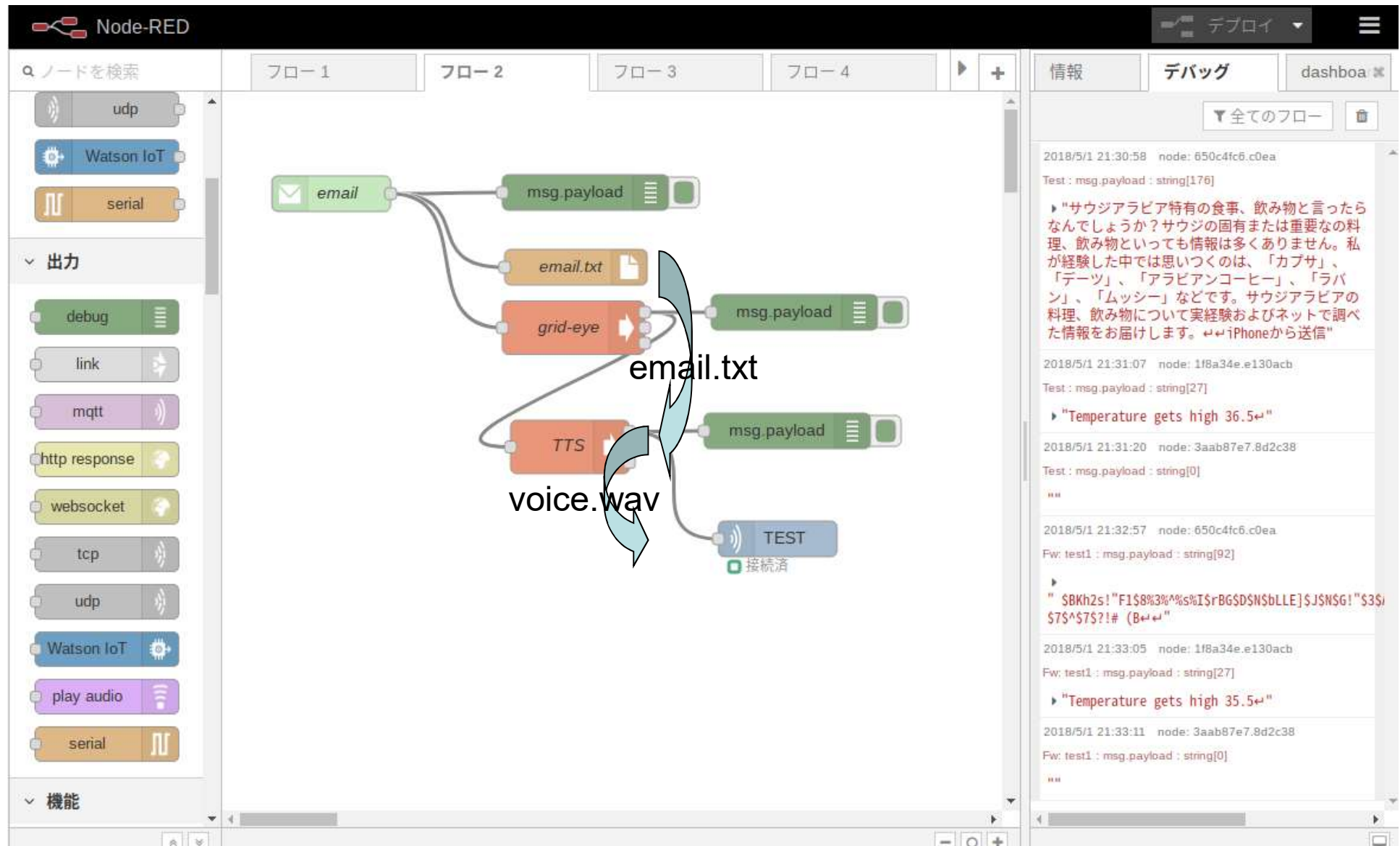
### はじめに

この記事は、[関西うちハックのイベント](#)にて試作した入館・入退室管理システムにて（以下の写真参照）、音声を出力するために日本語音声合成(Open JTalk)をRaspberry piで動作させるまでの方法をまとめたものです。



# できたこと

メールを受信して、温度35℃以上を検出したら、メール内容を読み上げる



# できなかったこと

- メールが複数着信していたときの処理
  - 最新のメールしか読み上げられない(古いメールは上書きされてしまう)
- メールがSJISだったときの処理
  - 文字化けしていて、そのまま読み上げられる
- メールに改行がはいったときの処理
  - OpenJTalkが最初の1行しか見てくれない
- 読み上げの不自然さ
  - 長文ではOpenJTalkが途中で急にしどろもどろになる
  - 文章理解や言い回しがいまいちで、高齢者に受け入れられる？

+\$BD9\$\$5Y\$, \$b\$"\$C\$H\$\$\$&4V\$K=\*\$o\$J" #F|\$+\$b\$`\$?\*\$E,v\$,O\$`\$b\$`\$9\$M#+(B  
+\$B,E,v%b<`\$X\$N0Z\$;49\$(\$O!`\$J\$+\$J\$+87\$7\$\$\$b\$N\$, \$`\$b\$`\$9\$!&!\*+(B  
+\$B\$O\$ \$-\$C\$F\$\$\$-\$7\$H,W\$\$\$`\$9!#+(B(`^)

# まとめ

- NodeRedはワンパス通すだけならめっちゃ速い
  - 細かい条件処理をいれていくと、どうなるかは不明
  - message.payloadわかりにくく、逃げたくなる
  - 良いNodeをライブラリから見つけてくるのが大切
- 生活ニーズにこたえるアプリが自分で簡単に作れたら、なんてすばらしいことでしょう ム(°▽°)ノ
  - しかし道は長そうだ・・・ (-。-) フゥー

ご清聴ありがとうございました